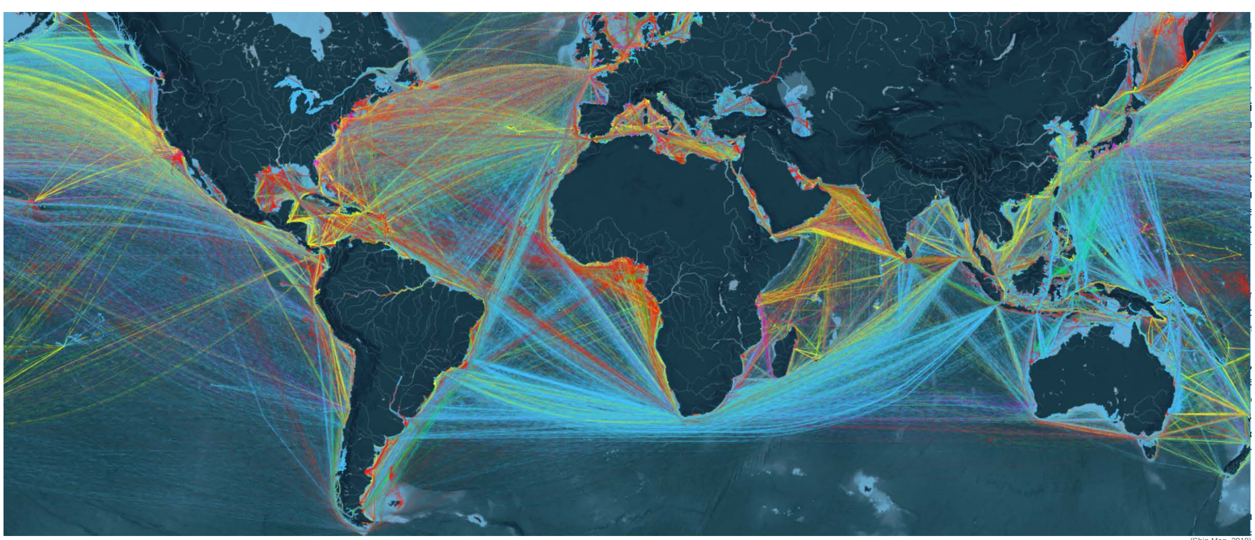
Portland State University

Maseeh College of Engineering and Computer Science

ETM
ENGINEERING & TECHNOLOGY MANAGEMENT

A Technology Roadmap to Uncontested Market Space using Autonomous Vehicles in the Transportation Industry

Course Title:	Capstone Project
Course Number:	ETM 506
Instructor:	Dr. Tugrul Daim
Term:	Summer
Year:	2018
Author:	Bobby Romanski



(Ship Map, 2018)

Table of Contents

Introduction

Overview

Rapid computerization of the sales, service, administrative, production, and transportation industries has given decision makers the ability to automate more of their workforce, such that high human unemployment, hyper competition due to increased demands on schedule and location flexibility, and a sharp decline in vehicle ownership to decrease the idle time of vehicles, will be untimely opportunities unless society is prepared for them. The only reasonable solution for this appears to be the creation of Autonomic Transportation Systems, which when paired with Autonomous Vehicles, will create Autonomous Transportation Businesses that employ Autonomous Vehicles for human and freight transportation. These vehicles include, aircraft, ocean freighters, rail, road and even space vehicles.

The Autonomous Transportation Business

Given that these vehicles have numerous well tested rules that apply to their administration and operation, adding autonomy to them may be as simple as installing sensors, engineering software, and enacting legislation. Thus, this paper will investigate how transportation vehicles can be granted individuation and the ability to conduct their own business operations. For example, after a car is assembled at a manufacturing plant, rather than being transported to a dealership, it inquires with the open market as to where its services are requested, purchases a ticket on an ocean freighter, and after a week at sea, docks on a different continent and drives to its contracted location of employment. During its life of service, it will order maintenance services, negotiate on the open market for contracts, and when it is near the end of its useful life, will provide free service to a contractor who will remove it from service via scrap or transform it into a new type of machine.

The Future Labor Market

Given the apparently imminent future of autonomous systems, it is predicted that business operations of the transportation industry will become autonomous, commoditized, and operate in hyper competition with very high barriers for human entry. The convenience, hidden complexity, and low cost of autonomous transportation may also facilitate fundamental changes in how humans interact with computers, giving rise to organic computing. Also, economic assistance may begin to be provided in the form of autonomous businesses rather than money and Human Governments will interact with Artificial Intelligence collectives to advocate for the needs of Humans and Autonomous Machines. This redirection of human resources from future industries of high computerization and unemployment to future industries of high employment demand and low computerization such as education, arts, health and STEM occupations, will allows humans to transition into work that provides a high quality of life, supported by autonomous industries.

Why Create a Technology Roadmap

A technology roadmap is a form of strategic communication (Phaal, & Muller, 2009, p.47) that, "…business and government leaders easily understand." (Galvin, 204, p.101) and improves collaboration among differing business units. Most importantly technology roadmaps help insure that products are ready for release at the proper time (Gerdsri, et all, 2010, p. 240).

Literature Review

Overview

Before creating decisions, the processing of incoming data is important, and this includes how to classify incoming problems (Xu, et all, 2018). Although cloud computing is popular nowadays for computation, edge computing is important for time sensitive applications, especially when there are requirements for location accuracy and mobility (Ahmed & Rehmani, 2017).

Once large volumes of data are gathered, processing it at the lowest cost and highest quality is important and this can be done through choosing "simplicity at the expense of accuracy and performance" (Chang, 2014). Another method to increase this performance is to partially process the data offline before streaming it, thus improving the overall quality of the final processing (Huan, et all, 2018). Since this data will eventually be stored and/or processed in a cloud, methods of adding and removing pseudo data such that it can be processed by shared machine learning services and its resultant analysis returned to the customer where the pseudo data is removed with the effective results of the machine learning processing being retained (Li, 2018). These self-regulating autonomic systems must also be able to optimize their decisions in the absence of central management and when faced with an information deficit (Pournaras, 2017).

Examples of this data include how to make traffic flow around large vehicles more efficient. For example, there is a 5% increase in the probability of a vehicle accident occurring when large vehicles constitute 30% of the traffic flow. This may be due to the increased number of lane changes when smaller cars maneuver around heavy trucks (Moridpout, 2015). As for aviation, over 30% of a passenger aircraft service's cost is computation and employee related (States News Service, 2012), thus cooperating autonomous systems may eliminate this cost.

Despite the time and cost savings of removing human labor from the transportation industry, the human ergonomics of autonomous systems are the most important benefit, thus the complexity of the system must remain hidden from human users (Cong, 2016). These ergonomics carry over into what is called the "Cyber-Physical Society" where the "…Cyber-Space, Physical Space, and Social Space…" are connected such that "super-links" are formed which enable cross-spatial relationships between nodes (Zhuge, 2014).

Communication

Efficiently coordinating the activities of networked machines requires communication systems engineered for the functions of these machines. In the case of today's vehicles, this lack of 'vehicle to vehicle' communication limits their cooperation (McCluskey, 2016, p.100). However when these communication channels are established they should be broadcast based rather than peer-to-peer (McCluskey, 2016, p.100), probably for the reason that broadcast will result in more data being instantaneously available to the entire ecosystem of vehicles. This may result in the removal of traffic lights (McCluskey, 2016, p.283), which would be deemed unnecessary if vehicles already possess the same data that a traffic light would have.

Concerning the broadcast of data, the conciseness of this data is exponentially increased by using a closed loop control mechanism (McCluskey, 2016, p.112) as shown in Figure 1. For example, loop (b) is an open loop, where changes detected by a sensor with input δ, are uploaded to (y). Being a sensor, most of this data is noise and thus irrelevant. In the closed loop (a), the changes (δ) are processed through the loop repeatedly until no reasonable changes are detected, thus relatively noiseless data is uploaded.

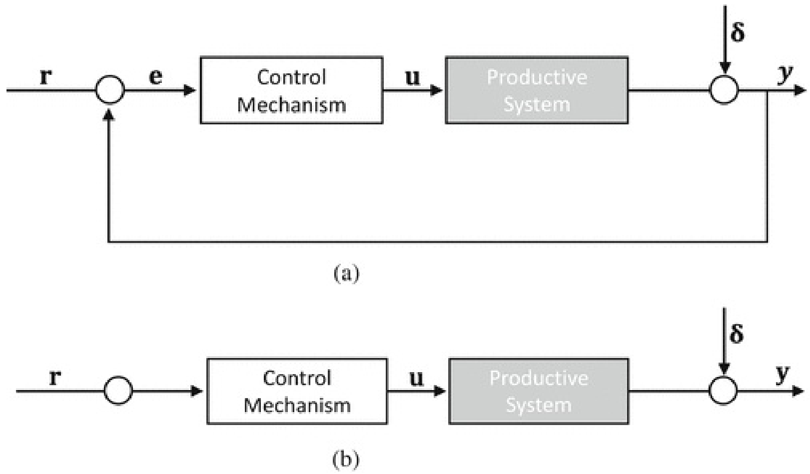


Figure 1 (McCluskey, 2016, p.190)

An extreme example of this is an open loop motion detection camera that uploads gigabytes of video footage to a server which then determines whether motion occurred, versus a closed loop camera that uses its own embedded video processor to determine whether motioned occurred and then uploads Kilobytes of decision data to a server. Thus, when potentially networking millions of vehicle communications channels in a metro area, closed loop control systems are preferred. This concept of efficient networking will be further elaborated under the concept of **Organic Network Control**, which optimizes and reconfigures network devices autonomously (Muller-Schloer & Tomforde, 2018, p.441) .

Autonomic Systems

IBM discovered in the early 2000s that the computation bottleneck is no longer a significant issue when compared with the bottleneck of systems integration (Kephart & Chess, 2003). Proposed solutions have been to research how systems can make intelligent decisions to administrate themselves without significant human intervention. This has led to the creation of the 'Four General Aspects of Autonomic Systems': Self-configuration, Self-optimization, Self-healing, and Self-protection, (Appendix A) (McCluskey, 2016, p.108) as well as 'Eight Conditions of an Autonomic System': self-aware, self-adaptive, self-optimizing, self-healing, self-protecting, self-interacting, with open standards, and anticipatory (Appendix B) (Muller-Schloer & Tomforde, 2018, p.542).

Given these aspects and conditions, the most foundational of them all is that "AS[Autonomous Systems] must be able to monitor its operational context as well as its internal state in order to be able to assess if its current operation serves its purpose" (Cong, 2016, p.140). This will be further explored in the product drivers of the technology roadmap.

Autonomous Freelancing in Transportation

First we must look at a vehicle that can make its own decisions. This includes, driving, searching for customers, maintenance, and recreating itself to be useful in a dynamic market i.e. relocating, software upgrades, mechanical upgrades, etc. Due to machines being able to generally outperform humans in jobs that do not require significant social intelligence, creativity, or perception and manipulation (Frey, & Osborne, 2017, p.31), such as driving, the competition that will be created as a result of autonomous vehicles competing for business with each other, will render humans unable to compete in this market.

To further illustrate this competition, we should observe that autonomous machines if designed correctly, should be able to return their utility to their normal rate, a.k.a. homeostasis, when subject to disturbances (Muller-Schloer & Thomforde, 2018, p.160), such as a traffic jam or a sudden change in their home base of operations, i.e. moving to an area thousands of miles away due to better market conditions that appeared there. We should also note that unlike humans, machines can reside in areas that are naturally or otherwise uninhabitable to humans such the ocean, underground, high angle landscapes, and far in the country side where no human support infrastructure exists, i.e. water, foodstuffs, or electricity. Examples of this include transporting the vehicle in a container and storing the vehicle in an inexpensive area while there is no business available to generate revenue. In the case of a human, this type of hibernation and relocation is not possible due to the need for food, oxygen, and other obligations outside of work.

If we are to define hibernation as being able to sustain oneself without generating revenue, simple tasks such as refueling or recharging the vehicle may also not be possible for humans. For example, if fuel and charge stations are used by autonomous vehicles that can wait in line for days without incurring expenses and to reduce expenses the number of these stations may be significantly reduced, then a human would be subject to waiting for hours or days without incurring any revenue. Although there is the alternative of scheduling an appointment for this, the human will certainly be charged more to fully fuel/recharge their vehicle, as that would disturb the efficiency of the autonomous vehicles that will partially refuel themselves in order to prevent the loss of revenue while refueling and save the station time so that the next car in line can receive their required amount of fuel. (McCluskey, 2016, p.263)

Good Enough Decisions

An algorithm can be defined as a set of steps that solves a problem. Improving an algorithms efficiency requires additional cost and design sophistication (McCluskey, 2016, p.10), thus to maximize overall utility of a system, a specification should be created that determines whether an algorithms answers are good enough so that unnecessary resources are not spent improving the algorithm unnecessarily. Thus there are cases where simple algorithms will be preferred to ones that are more accurate or faster (Chang, 2014, p.513). Thus there are cases where, "We are interested in 'good-enough' systems, i.e. systems that reach an acceptable state fast" (Muller-Schloer & Tomforde, 2018, p.156)

Organic Computing

Introduction

When autonomic systems relate with one another, a self-organized system begins to take shape, resulting in what can be called Organic Computing, described as Semi-autonomous agents that perform work locally and remotely, have large populations, interact with other agents, perform learning and optimization related to their work, the results of which are non-determined, and the final result is evolution of the total system. (Muller-Schloer & Tomforde, 2018, p.91) (Appendix C). It is interesting to note that this evolution requires large populations (Muller-Schloer & Tomforde, 2018, p.92), and it can be said that small populations will become uncompetitive due to their inability to optimize their work quickly due to a smaller sample size to work with.

The capabilities and benefits of Organic Computing are best utilized to "…increase the decision freedom of technical systems in terms of behavior and structure adaptation." (Muller-Schloer & Tomforde, 2018, p.186). Thus, the logic activity of this system is composed of three actions: trial & error, generalization, and programmed safety guarantees. (Muller-Schloer & Tomforde, 2018, p.192). For example, a stock trading program will check the results of its actions, determine how it can improve, and confine its actions to programmed boundaries, e.g. to stop trading if overall loss for the day exceeds 10%.

Since large populations are vital to the evolution of the system and since evolution is a result of the processes of the system, it can be said that "Processes are more important than the system elements" (Muller-Schloer & Tomforde, 2018, p.103). Thus, each element in the system must work at "finding the right balance between these two aspects—selfish autonomy and collective obedience" (Muller-Schloer & Tomforde, 2018, p.121) and when successful, the evolution and self-preservation of Organic Computing systems enables it to survive in the real world. (Muller-Schloer & Tomforde, 2018, p.28).

Given this vast collection of elements, Organic Computing appears to be analogous to Cellular Automata, in which, "The process requires large populations of interacting agents." (Muller-Schloer & Tomforde, 2018, p.89). Given that beneficial evolution is the desired goal of organic computing and cellular automata, their, "…behavior is mostly unintended and not explicitly designed" (Muller-Schloer & Tomforde, 2018, p.85). It is also interesting to note that this evolution cannot be explicitly programed because "… there exists no feasible method to derive rules or goals for the micro level such that a desired macro pattern results. The only way to do this seems to be by trial and error" (Muller-Schloer & Tomforde, 2018, p.176), thus automated experimentation is key and thus systems that are unexposed to a large population of other systems, will become obsolete due to their lack of evolution.

Groups of systems are defined as Holons, and when Holons connect with each other, interaction increases exponentially, and thus evolution, a.k.a. emergence, is achieved sooner. Organic Computing allows the complexity of these interactions to occur without significant human supervision, thus allowing additional functionality to be introduced into the system without the need for addressing the many fold increase in complexity. For in Information Technology systems a, "…25 % increase in functionality increases complexity by 100 %." (Muller-Schloer & Tomforde, 2018, p.31), thus Organic Computing reduces the need to address complexity before implementing new functionality. In the case of autonomous vehicles, new services, requirements, and requests can be added without much worry of their consequences since the system builds its own technical solutions (MULLER-SCHLOER, Thomforde, 2018, p.551).

This new way of solving problems requires a new way of thinking as shown in Appendix D (MULLER-SCHLOER, Thomforde, 2018, p.104). While autonomous computing systems are hierarchy centered, organic computing systems are holon, a.k.a. holarchy, centered (MULLER-SCHLOER, Thomforde, 2018, p.550). Given the choice of using either type of system, the choice for organic computing is clear when given tasks requiring a team effort as shown in Appendix E (Scholtes, et all, 2003, p.38). Given that teams are excellent at engineering solutions to problems, we can see that organic computing, which behaves like a team to change a system and "85% of problems can only be corrected by changing systems", otherwise known as the '85/15 rule' (Scholtes, et all, 2003, p.32).

Self Optimizing

As explained earlier, Organic Computer systems optimize themselves, which is advantageous because the system in effect designs itself while in production (MULLER-SCHLOER & Thomforde, 2018, p.34) rather than waiting for human engineers to design and test the system before it enters production. This is possible because “The system learns by remembering previous decisions and applying them repeatedly as long as possible.” (MULLER-SCHLOER, Thomforde, 2018, p.80). This is said to be, “…a yoyo design approach, where previous design decisions become subject to change” (MULLER-SCHLOER, Thomforde, 2018, p.286).

Some of these self-learned optimizations are undesired and can be avoided by fail-safes as mentioned earlier in this paper (McCluskey, 2016, p.15) which “…relieves system engineers from foreseeing all possible circumstances at runtime” (MULLER-SCHLOER, Thomforde, 2018, p.186). An example of this is the resonance catastrophe (MULLER-SCHLOER, Thomforde, 2018, p.48,51) where, for example, mechanical hard drives on a server rack all start and stop their heads at the same time in order to take advantage of RAID technology, causing catastrophic vibration such that none of the hard drives can read their data. Modern solutions for this behavior include algorithms to prevent all the heads from starting movement simultaneously.

Robustness

As mentioned earlier, an Organic Computing system must be able to heal itself after a disturbance and its ability to do so is defined as, “robustness [which] can be characterized by (I) the utility drop after the disturbance occurs and (II) the recovery gradient when the CM [control mechanism] is active” (MULLER-SCHLOER, Thomforde, 2018, p.160). Thus a system that does not show a high reduction of performance after a disturbance is said to be robust (MULLER-SCHLOER, Thomforde, 2018, p.92).

Emergence

What has been referred to earlier as evolution is named “Emergence” in organic computing. This is the creation of new knowledge from the process of Organic Computing systems interacting with each other. This is most efficiently done by an optimal combination of newness and affirmation in the communication relation between two partners.” (MULLER-SCHLOER, Thomforde, 2018, p.145). For example, a vehicular routing system and a popcorn machine have very little in common, however an ocean freighter and a crane at a destined port have will have plenty of new and useful information for each other. If we were to determine the most efficient way to unload a queue of ocean freighters with the crane, emergence would be required and these “emergent properties are not possible to be detected by microanalysis, they can only be observed by a holistic approach (MULLER-SCHLOER, Thomforde, 2018, p.102), thus an Organic Computing System can determine the best solution for this, rather than by analyzing a static list of cargo.

Trust

In addition to the privacy preserving techniques mentioned earlier which apply to distributed systems, we must also take into consideration the security of each element. For years, Trusted Platform Modules and most other chips believed to be secure, have been comprisable, given a reasonable amount of time and money (Skorobogatov, 2010).

Invasive & Semi-Invasive Decryption of Microchips

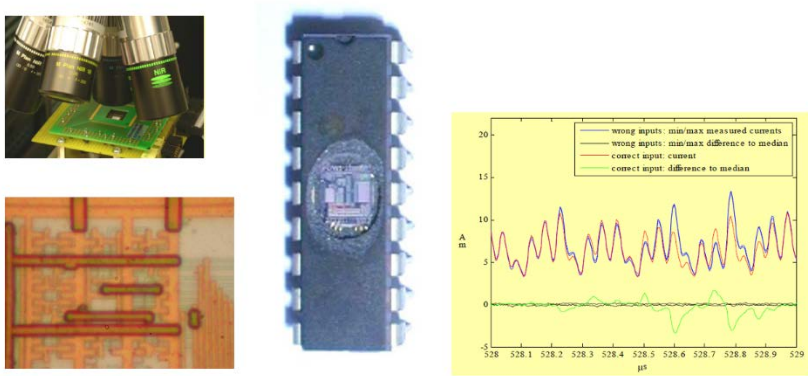


Figure 2 (Skorobogatov, 2010)

The author believes that a solution for this is to minimize the amount of unencrypted data stored in the computer’s memory, which can be done by distributing the data or storing it completely offsite where physical attacks to the computer components cannot be performed.

On the other hand, establishment of trust of a system of computers can best be implemented via design principals based on “Ostrom’s eight principles of Enduring Institutions” (Appendix F)) (MULLER-SCHLOER, Thomforde, 2018, p.100, 211). This will result in cooperative behavior which will lead to higher total efficiency in the system. (MULLER-SCHLOER, Thomforde, 2018, p.89)

Knowledge Management

When a computer system performs a knowledge transfer, some knowledge cannot be completely transferred and thus is referred to as “sticky information” (Von Hippel, 1994, p.4). For example, it is infeasible for a traffic light to download all of the traffic information in the region, however it can transfer small iterations of knowledge from a server concerning its road intersection and surrounding city blocks. Thus, this iterated knowledge transfer is less costly in throughput and processing requirements than a total knowledge transfer (Von Hippel, 1994, p.6). This can also be used as a form of security so that the transfer of knowledge can be regulated. (Von Hippel, 1994, p.10)

Spaces

Humans work in multiple spaces, such as cyberspace, physical space, and social space (Zhuge, 2014, p.180) . Establishing relationships between objects in various spaces facilitates interspace resource coordination (Zhuge, 2014, p.181), which creates value. Consider the following examples and how value can be derived from interrelating these spaces. For example, the concept of cars and traffic reside in different spaces, however the automated cars will benefit from knowing that they are in traffic and the people in the social space will benefit from knowing that there are many cars on the road, thus one can see the many opportunities for generating value by relating various spaces. Discovering these rules of interconnection is future research, which will create a coevolution of the spaces (Zhuge, 2014, p.180).

Physical space	social space
cars	traffic
art sculpture	idea & opinion exchange

Nature space	cyber space
trees	radio signal interference

social space	cyber space
festival	high broadcast video traffic

When these rules are coupled with Organic Computing, autonomous vehicles will be able to find the best customers & partners. For example, a basketball star in the social space, arriving within 50 miles of the vehicle's physical space, means the vehicle will have many customers who need transportation. Other examples include how, "An operation's motion energy can be measured by the number of individuals who have changed their communities and the total number of individuals" (Zhuge, 2014, p.182), thus giving an indicator of when an autonomous vehicle should reconsider the partnerships and methods it uses for generating revenue, such as when a major airline changes its hub to a different city.

This concept of multi-space coordination and optimization will enable computers to be represented as multiple artifacts & symbols in various spaces. For example, an autonomous vehicle can be represented as a car, guarantor, guardian, property owner, manager, etc. Also, if it is more profitable to be represented as a bus rather than a car, the vehicle may pay a mechanical service to transform it into a bus. Thus as spaces and their artifacts evolve, their corresponding business services also evolve.

Method

A literature review was performed on the topics of organic computing, autonomous transportation, anonymity, data privacy, bidding for business, dataflow, and vehicle traffic management. Business, market, product, and technology drivers were then created based on this literature. Drivers were then grouped together and placed on the roadmap as shown in appendix J, and every driver was assigned a code as shown in Appendix I. Relations between a layer and its upper layer are shown by using symbols and are explained in the Roadmap section.

Values were assigned to driver via the Quality Function Deployment (QFD) method with a score range of 1-9, with nine being the highest value. Each driver in the roadmap was scored by assigning a value to each of its relationships with drivers in its upper layer and then multiplying these by the weight of these upper relationships as determined by their average score in their upper layer.

The relationships shown in Appendix I were determined by the knowledge of the author and whether the relationship of the driver has a score of 7 or higher in the QFD tables. Effort was also placed in trying not to associate each driver with more than three upper layer drivers. The timeline assigned to the drivers were based on industry experience, literature search, and whether drivers in lower layers can support their timely deployment.

Roadmap (see Appendix J)

Symbols

A driver's connection to its upper layer is shown by matching its symbol on the left, with its related drivers in the upper layer that have its symbol on their right side. The driver codes for example T1 for 'Quantitative Emergence' as shown in Figure 3, will be referred to in the following sections. For example when speaking of a technology involving Quantitative Emergence, it may include its code (T1) which refers the reader to check this code in Appendix H to see which driver is being referring to.

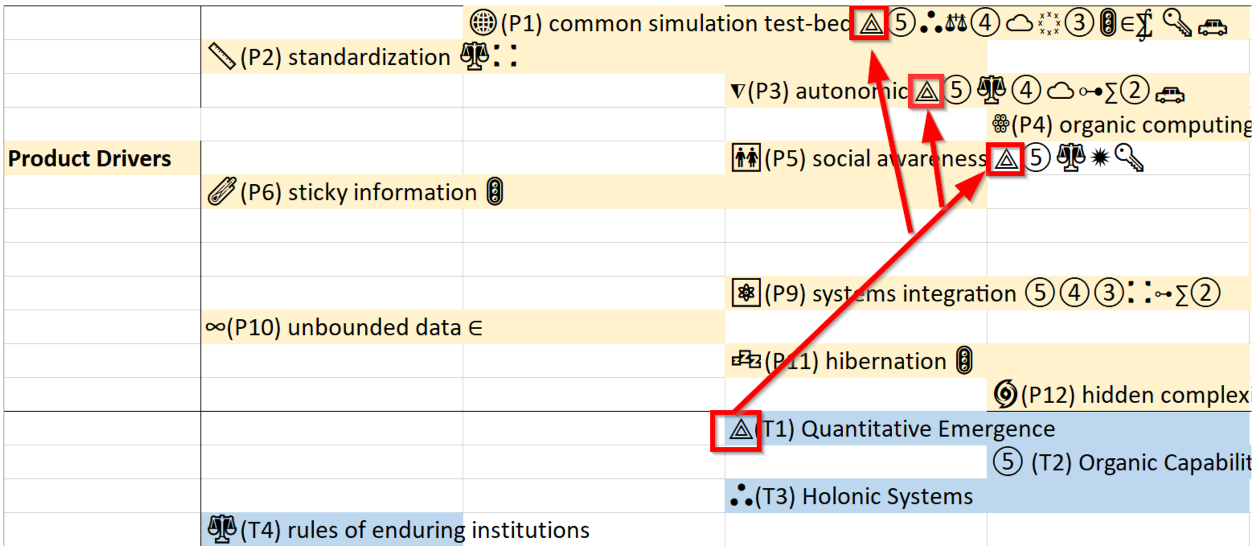


Figure 3 Connections to Upper Layers

Drivers

Overview of Business Drivers

Blockchain enabled smart contracts, diverting manpower into profitable businesses, and lack of monopolistic legislation in certain jurisdictions will make it easier to build businesses more quickly. This will make it easier for economies with low financial leverage to enter the transportation commodity market. It will also result in the automation jobs that do not require significant social intelligence, creativity or perception and manipulation (Frey, & Osborne, 2017, p.31). Thus it is recommended to invest in staff development and long-term retention of management, and jobs related to STEM, social service, education, and arts, due to the low probability of computerization of these industries. (Romanski, 2017).

Since the transportation industry has a high probability of computerization (Frey, & Osborne, 2017, p.40), human unemployment in this field will be high, thus Hyper Competition among humans and machines is inevitable, resulting in difficult working conditions for humans. Using organic computing, these industries' data will become deeply connected with competitors and to break out of the unprofitable cycle of Hyper Competition that will follow, it is assumed that a knowledge intensive business climate will emerge which will create profitable specializations.

Code ***Business Driver Name***
B1 *Fast economic development*
B2 *Hyper competition*

*B3	Deeply connected*
*B4	Standardization*
*B5	Knowledge intensive*
*B6	Computerization*

Market Drivers

### *M1	communication*

As mentioned in the literature, communication primarily refers to machine to machine communication, ranging from 'vehicle to vehicle' communication, to global communication as it relates to the transportation industry. This communication will preferably be done via a closed-loop control mechanism that broadcasts its data.

### *M2	location flexibility*

As communication and other supporting infrastructure for autonomous vehicles improves, they will be able to provide services that humans cannot, such as nonstop transportation over thousands of miles, storage in inhospitable environments, and the ability to cross borders without a VISA.

### *M3	automated decisions*

Machines can think like humans by using algorithms, and thus can provide services to humans given the right programming. There is already a demand for transportation related machines that can make their own decisions and there are plans to, "…increase the decision freedom of technical systems in terms of behavior and structure adaptation. (Muller-Schloer & Tomforde, 2018, p.186).

### *M4	time flexibility*

This is similar the 'M2 location flexibility' market driver, however it also infers that autonomous vehicles can work on nearly a 24/7 schedule, given time for maintenance and refueling. Also, this refueling time may be significantly reduced by adding additional batteries and fuel capacity as the market demands.

### *M5	adaptation*

Vehicles will need to respond efficiently to changing market conditions, ranging from traffic to a massive relocation of customers to different area. The computational capacity and complexity required for this may be enormous.

### *M6	trust*

Although it is assumed that individual endpoints are probably secure, the ability for an endpoint to abuse its privileges by using unfair practices such as malicious bidding or keeping secure client data for later use, will endanger the trust that other people and machines have in an autonomous vehicle management system.

### *M7	cross-space coordination*

This is the ability for an autonomous vehicle intelligence to establish links between spaces (Zhuge, 2014, p.181), e.g. Cyber-Space and Physical space, so that value can be created. This will create a, "super-link network, [where] a node in any space can link to any node in any space," (Zhuge, 2014, p.181). This is a knowledge intensive market demand and is valuable as demonstrated by the observation that "…knowledge-intensive companies around the world are valued at three to eight times their financial capital." (Becerra-Fernandez & Sabherwal, 2015, p.15).

M8 privacy

Certain levels of privacy are standard practice for the transportation industry. The ability to comply with wider scopes of privacy will provide opportunities in more markets.

M9 computation

Vehicle routing requires the use of algorithms, some of which will require a massive amount of computation when scaled for the use of real-time regional traffic planning. This scale of computation may not be available onboard the vehicle and thus will have to be offloaded to a data center or mesh network.

M10 lower cost

Cost is a measure of efficiency, however time is usually more expensive than money in today's market. It may be predicted that as the market for automated vehicle services increases, costs will decrease.

Product Drivers

P1 common simulation test-bed

The product and technology drivers require testing, and if the tests can be repeated and scaled, then standards can be created. More specifically, a common simulation test-bed will accelerate development of autonomous (M3) and organic control mechanisms (McCluskey, 2016, p.15), resulting in enhanced adaptation (M5), more efficient computation (M9), and the ability to do further research in cross-space coordination (M7).

P2 standardization

Communication (M1) and privacy (M8) standards have existed for decades and can be applied to new technologies, however, implementation of future research in cross-space coordination (M7) will require new standards. Also, "Standardization prevents problems from reoccurring." (Scholtes, et all, 2003, p.31), and lowers costs (M10).

P3 autonomic

An autonomic product answers the market need for automated decisions (M3), and thus being automated, can be programmed for use at given flexible times (M4) , as well as the ability to adapt (M5) to basic changes such as rerouting for traffic when a given threshold is reached.

P4 organic computing

Organic Computing uses holarchical (holon-centric) communication (M1) and a robust, self-optimizing (M3, M5) processing system that facilitates the communication of holons. These holons can facilitate the communication of cross-space communication (M7)

P5 social awareness

A socially aware computer will restrain itself (M3, M6) from harmful behavior such as, "Rational behavior without social awareness [which] leads to suboptimal utility of the individuals and the collective" (MULLER-SCHLOER, Thomforde, 2018, p.206). e.g., requesting routing instructions from a cloud service so often that it causes a denial of services to other clients.

P6 sticky information

The ability to transfer knowledge iteratively gives the autonomous vehicle the freedom of retrieving knowledge remotely (M2), allows the knowledge owner to regulate (M6) access to their knowledge, and also allows unrelated data to not be transferred, thus maintaining privacy (M8)

P7 emergence

Emergence is a useful discovery resulting from numerous interacting autonomous (M3) and organic decision makers that have adapted (M5) to their environment. The value of this emergence is significantly increased when it concerns cross-space activity (M7).

P8 equilibrium systems

Equilibrium focused design improves stability. "Most natural systems are in a state of equilibrium or try to achieve such a state. Technical systems will be more stable and function more reliably if they are designed as equilibrium systems" (MULLER-SCHLOER, Thomforde, 2018, p.102). However, research is still pending on creating ways to develop and interact with equilibrium systems (MULLER-SCHLOER, Thomforde, 2018, p.102**.** Successful development of this will allow autonomous systems (M3, M5) to interact with an equilibrium system using methods of Computation (M9) that are beginning to be discovered.

P9 systems integration

This is the performance bottleneck of today's information technology (Kephart & Chess, 2003) and resolving it via organic computing will grant the ability to automatically upgrade and rearrange entire systems without the significant need for human assistance. Also, it will enable, "…integration between online learning and offline learning" [(M2,M7)] [which] is achieved through global monitoring and pub/sub model update[see T12]" (Huang, et all, 2018, p.6). Integration will also significantly lower costs (M10).

P10 unbounded data

The ability to process incoming data will give an automated decision system (M3) the ability to work with the most recent data, thus giving it the ability to work with updated information consistently (M4) and make necessary adjustment (M5).

P11 hibernation

This is the ability to stop all value generating functions and remain ready to resume these functions, which ultimately lowers the cost of service (M10). Examples include storage of a vehicle inside a container, waiting in line for hours or days (M4) for refueling, or stopping a vehicle mid route for months while waiting for snow to melt from the road (M2, M5).

P12 hidden complexity

Complexity must be addressed before functionality is added to a system. For example, every 25% increase in functionality in an Information Technology system, there is a 100% increase in complexity (Muller-Schloer & Tomforde, 2018, p.31). This can be addressed by firstly, recognizing the state of evolution that a system or product is in. Experiments on cellular automata have found four stages of complexity (MULLER-SCHLOER, Thomforde, 2018, p.81):

1. A stable homogeneity
2. Stable homogeneity with some oscillating patterns
3. A plethora of chaotic patterns
4. Stability made up of complexly interacting patterns which take a long time to create

Hiding this complexity from users and machines, probably through layers of abstraction, will enable them to understand the system more easily (M3, M7) and thus more easily exploit the system to create value.

Technologies

T1 Quantitative Emergence

Emergence (P7) is a product of an organic computing system(P4). The ability to quantify emergence results in the ability to manage and test it (P1). The scope of this testing may include autonomics (P3), measuring the fairness of a system (P5), and complexity (P12)

T2 "Organic Capabilities C5 (Adaptive with offline rule generation and cooperation)"
This is the pinnacle of organic computing and embraces all product and market drivers. It can: Adapt, learn online and offline, and cooperate with other systems to achieve a common goal.

T3 Holonic Systems

This is a technical imitation of a biological life system by interrelating organic computing systems (P4, P7, P12), and is engineered to imitate life because, "Life systems show us how to create emergent value from layers of semi-autonomous systems (Muller-Schloer & Tomforde, 2018, p.8)" This allows a system to organize and act upon massive amounts of complexity (MULLER-SCHLOER, Thomforde, 2018, p.100) by creating higher level abstractions from lower level agents (MULLER-SCHLOER, Thomforde, 2018, p.120)

T4 Rules of Enduring Institutions

Autonomous (P3) and Organic Computing (P7) agents will be programmed to use these rules (P2) (Appendix F) to conduct cooperative (P5) transactions because, "co-operative behavior leads to globally efficient patterns" (MULLER-SCHLOER, Thomforde, 2018, p.89)

T5 Evolutionary Simulation

These simulations (P1) will help quantify the effectiveness and value and costs of organic transactions, emergence, systems integration, and complexity.

T6 "Organic Capabilities C4 (Adaptive with offline rule generation)"

Unlike the pinnacle of organic Computing (T2), this system has no ability to cooperate with other organic computing systems, and thus lacks social awareness (P5).

T7 Cloud Computing

Cloud computing will provide autonomous (P3) and organic (P4) computing systems with an established albeit relatively slow, line of communication when compared to edge computing or mesh networks.

T8 Swarm Robotics

Using Swarm Intelligence (P4) (Muller-Schloer & Tomforde, 2018, p.512), groups of robots can be tasked with cooperation in a scalable way that is, flexible, and robust (P12) (Muller-Schloer & Tomforde, 2018, p.505).

T9 "Organic Capabilities C3 (Adaptive with online learning):"

Unlike the pinnacle of organic Computing (T2), this lacks social awareness and ability to function offline or in environments where the knowledge available online lacks rich content.

T10 TIMIPLAN

TIMIPLAN is a software application that uses Linear Programming and Automated Planning Techniques to solve multi-modal transportation problems. For example, evaluating a scenario of driving cargo to a dock, loading it on a ship, and then adding X additional routes and modes of transportation to the scenario (McCluskey, 2016, p.299). It can also be used as a traffic management planning resource for monitoring, adjusting, and providing recommendations to vehicles in traffic (P6, P8, P12) (McCluskey, 2016, p.314). It is used in combination with PAUSE(Progressive Adaptive User Selection Environment), which creates a combinatorial auction environment where automated vehicles can bid for work.

T11 Open distributed systems

Some functions require the use of multiple computers to solve a problem. For example, knowledge of traffic flow will require data from multiple sensors. The ability for these systems to communicate will require standardization (P2) and integration (P9)

T12 data flow model

In the transportation industry, the best routing solutions change as time passes, therefore there is no feasible chronological boundary for the data required to solve these problems. However, the data must be processed to provide an answer at a given time, and the techniques of the data flow model provide this. (Akidau, et all, 2015)

T13 Machine learning

This enables computers to dynamically create statistical models to solve problems based on the data presented at production time, rather than use a static algorithm created at design time.

T14 Organic Network Control

This allows a network to self-adapt and self-optimize based on dynamic conditions. Self-adaption enables two features: mode selection and peer to peer.

Mode selection is the ability to change the network medium and periodic behavior (Muller-Schloer & Tomforde, 2018, p.442) . For example, when a vehicle is out of range of a cell phone tower, it will change to a low bandwidth satellite communication (network medium) and therefore will refresh its data more sparsely to reduce throughput (periodic behavior).

Peer to Peer utilizes peer to peer protocols instead of client-server protocols. (Muller-Schloer & Tomforde, 2018, p.446)

As for self-optimization, this is done by using techniques to optimize throughput (Muller-Schloer & Tomforde, 2018, p.440) and managing the self-learning delay (Muller-Schloer & Tomforde, 2018, p.448). The self-learning delay determines when to adapt to a circumstance, for example, whether to process alternative driving routes periodically or only when the vehicle is off route.

T15 Quantitative Autonomy

The is the ability to process information without having to offloading tasks to a separate system (P3). For example, the ability for an autonomous vehicle (P3) to create its own route, rather than having to obtain the route from a server. Being able to autonomously share information via an organic computing system will lead to emergence (P7) and creation of an equilibrium system (P8).

T16 Trust Communities

Implicit trust communities allow members to select which who is trustworthy (Muller-Schloer & Tomforde, 2018, p.481) while Normative trust communities centrally dictate which members are trustworthy (Muller-Schloer & Tomforde, 2018, p.486). The recommended design principles of these are illustrated by the principals of enduring institutions (see appendix F) (MULLER-SCHLOER, Thomforde, 2018, p.100, 211).

Interaction outside the trust community can use privacy preserving technologies such as anonymous queries that use blockchain as a search mechanism, a.k.a oblivious keyword searching (Jiang, Guo, et. all 2017). Another alternative method for untrusted agents is to allow them to search encrypted data using keywords that have been pre-authorized by the host (Jiang, Mu, et. all 2017).

T17 "Organic Capabilities C2 (Non-learning adaptive):"

Unlike the pinnacle of organic Computing (T2), this lacks social awareness, the ability to function offline or in environments where the knowledge available online lacks rich content, and is its intelligence is limited to static algorithms rather machine learning methods that use statistics to adapt to new situations.

T18 PLANETS

This is a traffic simulator that can be used for decision making such as which lane to use, or which street to turn on (McCluskey, 2016, p.89). Its function is primarily informational as, "global control strategy is provided from a Traffic Management Centre, but traffic participants have a freedom to make decisions autonomously" (McCluskey, 2016, p.23).

PLANETS is divided into six applications:

"– Learning App—updates a model of travel times for different routes based on local history and information provided by the TMC [Traffic Management System]
– Routing App—provides access to external (or vehicle-internal) routing services
– Grouping App—implements group formation protocols and corresponding decisions
– CommBox—provides access to basic 'Vehicle to Vehicle' and 'Vehicle to Internet' functionality
– setRoute App—translates the vehicle's tactical plans (route) to corresponding operational actions" (McCluskey, 2016, p.90)

T19 "Organic Capabilities C1 (OC-ready)"

This has no Organic Capabilities functionality whatsoever, however it's hardware contains the potential of being programmed to have these capabilities and thus it promotes this capability and can be placed into existing infrastructure so that it can be activated when the technology is ready.

Analysis

Introduction

Based on the results of the roadmap, the following interpretation for each layer have been made. A synthesis of these predictions are presented in the Conclusion of this paper.

Business Environment Predictions (business drivers)

Standardization and Computerization of business processes continues from previous years. Deeply connecting organizations begins, which then allows for fast economic development. When development reaches a plateau, hyper competition begins and the only escape from this will be a breakthrough in knowledge, requiring operations to become knowledge intensive.

Market Predictions (market drivers)

There is continuing development of inexpensive machine to machine communication that facilitates the exchange of vital data. This data will enable machines to make good business decisions and execute them in various spaces, the results of which we have yet to imagine.

Product Predictions

Standards will continue to be created and implemented, enabling new autonomic and organic computing processes to be tested and integrated with other systems. After these systems are released for widespread use, they will reveal emergent patterns that humans will utilize to create value by connecting these patterns to other spaces.

Technology Predictions

None of these technologies are possible without cheap and instantaneous machine to machine communication. Standard technologies from the beginning of the 21st century will serve as a low level communication and decision making layer for Holonic Systems and Organic Computing.

Conclusion

Distinct Phases

The Business and Market Driver layers show three distinct phases: integration, autonomous decisions, and discovery. The integration phase, occurs until 2030 and improves standards and communication using well known technologies so that data can be processed and trusted.

The autonomous decisions phase occurs from 2030 to 2038 and can only occur because the lower management and coordination level of the transportation industry have been computerized and interlinked with other related transportation systems, thus the pace of business and competition will vastly increase however autonomous decision making will be handling most of this work using Organic Computing Systems. However, these decisions will not be possible without the ability to simulate them in a virtual laboratory (P1) and utilization of the communication infrastructure from the prior phase will serve as a medium for this 'autonomous decisions' phase.

The discovery phase from 2038 to 2050 demands new knowledge from the autonomous decisions phase such that hypercompetition will end because of a plethora of new value creation opportunities based on relating objects from different spaces (M7)

Preparing for the Discovery Phase

Given that a lucrative future of value creation in uncontested market spaces is due during the discovery phase, it is best to prepare for this phase now, by specializing ones business in objects that reside in spaces of ones choosing and finding ways that these objects can relate with one another in ways that create value, as shown in the examples in the Spaces section.

It is also interesting to note that the Common Simulation Test-Bed (P1) received exceptionally high scores, as none of the other drivers can create decisive information without it. Thus, an organization preparing for this phase may wish to hone its abilities in Computational Science so that they can simulate experiments and thus create their own Simulation Test-Bed. Device manufacturers should also determine which devices in their future production environment will handle the majority of their computation. For example, if the market demands that most of the simulation processing is done via an edge network, then devices that can handle the required communication, collaboration, and processing must be installed in the vehicles, road infrastructure, smartphones, etc. , rather than on a stack of expensive cloud servers that lag significantly behind the edge network.

Further Research

Edge Computing

Edge computing is a product driver that was accidentally not included in the roadmap and if time permits, the roadmap should reevaluated to see how this driver will affect it. It is a unique driver in that it decreases network latency and allows access to distributed systems when cloud computing is inaccessible (Roman et. all, 2018, p.1). Edge computing is also superior to cloud services when requesting "…local contextual information, such as precise user location, local network conditions…" (Roman, et all, 2018, p.60) that only local devices will have the ability to sense. This will enhance the user experience of "delay-sensitive applications, such as vehicular networks and augmented reality" (Roman et. all, 2018).

Edge computing also provides another means to create a closed loop as mentioned in the Communication section such that local devices can communicate with one another to create a decision and upload these results to a cloud server, thus reducing noise and throughput (MULLER-SCHLOER, Thomforde, 2018, p.91). This also applies to business processes that such as smart contracts, which can be completed on an Edge network for upload to a cloud server later (Prybila, et. all, 2017), such as a delivery of cargo to a seaport that does not have network access.

It is also important to note that centralized computation is not scalable due to lag and processing requirements (MULLER-SCHLOER, Thomforde, 2018, p.422). For example, it is more economical to process information on thousands of idle customer smartphones, than it is to build and manage a stack of redundant cloud servers, as well as collocating those servers close to the customers to reduce network lag time. Rather, if the customer's smartphones can communicate with each other on an edge network to make a decision, then lag is significantly reduced.

Quantification

The quantification of emergence and processing times of organic functions will be necessary to determine how long a process/decision will require. This will resolve the problem of transaction confirmation times (MULLER-SCHLOER, Thomforde, 2018, p.15), which is critical because without it, one will not know if a process will take, for example, seconds or months to complete, as demonstrated by the concept of Big O in computer science.

Although the quantification of these concepts must occur, creating a roadmap of how this quantification can be discovered will be beneficial, as this technology is due by the year 2030 and, "…quantitative analysis of technical self-organising systems is still on-going research" (MULLER-SCHLOER, Thomforde, 2018, p.137) and, “emergence quantification is important for risk controls and determining responses“ (MULLER-SCHLOER, Thomforde, 2018, p.137). For example, if we can quantify the effectiveness of a control mechanism, then we can begin to quantify how robust it is, resulting in the ability to determine how resilient a system is when disturbances occur (MULLER-SCHLOER, Thomforde, 2018, p.161).

Emergence Centered Management Methods

The decision making tools that use static algorithms, mechanisms, and models may be replaced by tools that primarily use live-data to and emergence solutions from Organic Computing systems to create a decision. Surprisingly, these organic systems may be quite controllable as it was found that, “…sparse inhomogeneous networks were difficult to control whereas dense homogeneous ones could (only) be controlled via relatively few nodes, which, interestingly, were not the high-degree nodes.” (MULLER-SCHLOER, Thomforde, 2018, p.113). Therefore we can see that organic networks or ‘Holons’ do have control mechanisms which are in the form of nodes as mentioned above.

This also brings into question whether only dense homogenous network will produce the best value, and thus sparse networks may have little if any value, as controlling them and obtaining emergence from them is difficult due to their low population count. Thus in the future, enterprise systems will probably be managed from the bottom-up via organic computing, rather than the top-down via pre-determined control mechanisms.

Appendices

Appendix A

Four General Aspects of Autonomic Systems
(McCluskey, 2016, p.108)

Concept	Current computing	Autonomic computing
Self-configuration	Corporate data centres have multiple vendors and platforms. Installing, configuring and integrating systems are time consuming and error prone	Automated configuration of components and systems follows high-level policies. The rest of the system adjusts automatically and seamlessly
Self-optimization	Systems have hundreds of manually set, non- linear tuning parameters, and their number increases with each release	Components and systems continually seek opportunities to improve their own performance and efficiency
Self-healing	Problem determination in large, complex systems can take weeks for a team of programmers	The system automatically detects, diagnoses and repairs localized software and hardware problems
Self- protection	Detection of and recovery from attacks and cascading failures are manual	The system automatically defends against malicious attacks or cascading failures. It uses early warning to anticipate and prevent system- wide failures

Appendix B

"…eight conditions that have to be fulfilled in order to call a system 'autonomic'."

(Muller-Schloer & Tomforde, 2018, p.542)

1. The self-managed system has to be self-aware in terms of knowing (i) its own resources it has access to, (ii) its own capabilities and limitations, and (iii) its relations to other systems in its vicinity.
2. The self-managed system is self-adaptive in terms of changing parameter configurations in response to dynamic environmental conditions.
3. The self-managed system is self-optimising in terms of improving a given system utility over time.
4. The self-managed system is self-healing in terms of managing occurring problems and repairing itself (or circumvent the problematic area).
5. The self-managed system is able to identify possible attack vectors and autonomously encounter them.
6. The self-managed system has to interact with other systems in its neighbourhood to establish and maintain communication.
7. The self-managed system has to be based on open standards.
8. The self-managed system is anticipatory with respect to the demands for its resources and simultaneously hides these resources to users.

Appendix C

10 Characteristics of Organic Computing
(Muller-Schloer & Tomforde, 2018, p.91)

1. **Semi-autonomous Agents** Systems consist of active units that have a certain awareness of their environment (by observation) and can act in this environment. For this, they have to take decisions. They are autonomous only to a certain degree because for their decisions they take external and internal constraints into account. This property is called 'semi-autonomy'. Independent units—technical or natural—are called 'agents'.
2. **Locality and Decentralisation** Agents observe their environment through sensors and act in and upon this environment through actuators. The sensory horizon and the sphere of action are predominantly local. This leads to decentralised systems. Decentralisation, however, does not preclude occasional coordination by possibly temporary superordinate agents with coordinating tasks.
3. **Large Populations** OC systems consist of (very) large numbers of agents. Some of the observed effects such as emergence will more likely occur in large populations than between only a few agents.
4. **Interaction** In all systems of interest in the OC context, the constituent agents interact with each other. This is not necessarily a fully meshed interaction topology but the interaction graph will always be connected.
5. **Learning and Optimisation** Agents learn at runtime in order to adapt to new situations. Learning presupposes an objective, i.e. a method to weight different action options against each other and in relation to the set objective. Learning according to an objective function guides the system into a desired direction. OC agents as single entities and the whole OC system as well try to optimise their performance.
6. **Non-determinism** Learning and optimisation explore an unknown fitness landscape. They perform trial-and-error moves in this landscape with a certain exploitative component (hill-climbing) but also with random movements. Moreover, the sensory input of the agents will be noisy, and the result of the agents' actions will depend on unknown and not controllable external influences. The result is non-deterministic behaviour.
7. **Evolution** Learning can take place within an individual agent but also within the population. Evolutionary methods allow for a directed (by the objective function) andat the same time non-deterministic trial-and-error exploration of the fitness landscape. Evolutionary learning requires large populations. Evolutionary processes play a role between the agents but can also be used within the "brain" of a single agent. In this case, evolution is simulated by virtual search agents that sample and evaluate the fitness landscape. This method is used by Genetic and Evolutionary Algorithms.
8. **Emergence** The interaction of large populations of individual agents with local behaviour can lead to macro patterns in space and/or time. The appearance of such macro patterns is usually not predictable (at least in detail). This effect is called 'emergence'.
9. **Self-organisation** Single adaptive agents or populations of interacting agents can change their structure or behaviour without explicit control from the outside world. Self-organisation is closely related to autonomy.
10. **Robustness** OC systems try to optimise their behaviour despite the presence of external disturbances. A system that does not show a (heavily) degraded performance when it is disturbed is called robust. More realistically, robust systems might degrade from a target performance but return to an acceptable behaviour within a short recovery time. While the other nine characteristics are descriptive for an OC system, robustness can be defined as their ultimate goal.

Appendix D

“Transition from traditional to systems thinking”
(MULLER-SCHLOER, Thomforde, 2018, p.104)

Old	New
Closed systems	Open systems
Rigid, brittle systems	Adaptive systems
Separated objects (atomistic)	Related objects, combined into systems
Single science	Interdisciplinary approaches
Object-oriented	Process-oriented
Mechanistic	Organic, cybernetic
Analytic	Integrated (synthetic)
Linear functional chains	Networked process loops
Exact mono-causal approach (with implications for the whole system neglected)	Interdependent structure with fuzzy parameters, conditional
Hierarchy, unilateral power relationships	Network, holarchy, bi-lateral power
Certainty, controllability	Non-determinism
Descartes, Newton	Alfred N. Whitehead, Schopenhauer

Appendix E

A List of Situations where Teams Outperform Individuals
(Scholtes, et all, 2003, p.38)

- The task is complex.
- Creativity is needed.
- The path forward is unclear.
- More efficient use of resources is required.
- Fast learning is necessary.
- High commitment is desirable.
- Cooperation is essential to implementation.
- Members have a stake in the outcome.
- The task or process involved is cross-functional.
- No individual has sufficient knowledge to solve the problem.

Appendix F

“Ostrom’s eight principles of Enduring Institutions”
(MULLER-SCHLOER, Thomforde, 2018, p.100, 211)

1. *Clearly defined boundaries: Those who have rights or entitlement to appropriate resources from the CPR are clearly defined, as are its boundaries.* An example of this principle regarding the Tragedy of the Commons (TtC) is that the part of land belonging to each peasant must be clearly defined. The same is true for the peasants eligible to use it.
2. *Congruence between appropriation and provision rules and the state of the prevailing local environment:* The rules must prevent overuse or degradation of the common goods.
3. *Collective choice arrangements: In particular, those affected by the operational rules participate in the selection and modification of those rules.* Regarding the TtC example, this means that the peasants farming the land also administer the rules defining the farming. This principle prevents third parties imposing their interests.
4. *Monitoring of both state conditions and appropriator behaviour is by appointed agencies, who are either accountable to the resource appropriators or are appropriators themselves.* This principle means that only such people may monitor the CPR who are involved in the CPR themselves. This prevents corruption and manipulated monitoring.
5. *A flexible scale of graduated sanctions for resource appropriators who violate communal rules.* In the TtC example this principle defines in which way a peasant violating the rules of farming can be sanctioned.
6. *Access to fast, cheap conflict resolution mechanisms.* A result of this principle is that the reaction to conflicts can occur fast, by e.g. changing the rules of farming or sanctioning a peasant.
7. *Existence of and control over their own institutions is not challenged by external authorities.* This rule states that the Enduring Institution must be self-ruling. External authorities overriding the rules might endanger the stability of the system.
8. *Systems of systems: CPRs can be layered or encapsulated.* This principle means that hierarchies of CPRs are possible in order

Appendix H

	Product Drivers				Business Drivers
Code	**Description**	**Related Market Driver**	**Symbol**	**Code**	**Description**
P1	common simulation test-bed	M5, M3, M9, M7		B1	Fast economic development
P2	standardization	M1, M8, M7, M10		B2	Hypercompetition
P3	autonomic	M4, M3, M5		B3	Deeply connected
P4	organic computing	M5, M3, M7, M1		B4	Standardization
P5	social awareness	M3, M6		B5	Knowledge intensive
P6	sticky information	M2, M8, M6		B6	Computerization
P7	emergence	M5, M3, M7			
P8	equilibrium systems	M5, M3, M9			
P9	systems integration	M7, M2, M10			
P10	unbounded data	M5, M3, M4	∞		
P11	hibernation	M5, M2, M10, M4			
P12	hidden complexity	M3, M7			
	Technology Drivers				**Market Drivers**
Code	**Description**	**Related Product Driver**	**Symbol**	**Code**	**Description**
T1	Quantitative Emergence	P1, P3, P4, P5, P7, P8, P12		M1	communication
T2	Organic Capabilities C5 (Adaptive with offline rule generation and cooperation)	P3, P1, P7, P8, P12, P4, P5, P9	⑤	M2	location flexibility
T3	Holonic Systems	P1, P7, P12, P4		M3	automated decisions
T4	rules of enduring institutions	P3, P1, P7, P8, P5, P2		M4	time flexibility
T5	Evolutionary simulation	P1, P7, P8, P4, P5	✷	M5	adaptation
T6	Organic Capabilities C4 (Adaptive with offline rule generation)	P3, P1, P7, P8, P12, P4, P9	④	M6	trust
T7	Cloud Computing	p1, p3, p4		M7	cross-space coordination
T8	Swarm Robotics	p1, p12, p4		M8	privacy
T9	Organic Capabilities C3 (Adaptive with online learning):	P1, P8, P12, P4, P9	③	M9	computation
T10	TIMIPLAN	P1, P12, P8, P11, P6		M10	lower cost
T11	Open distributed systems	P2, P9			
T12	data flow model	P1, P10			
T13	Machine learning	P1, P4, P12, P8			
T14	Organic Network Control	P3, P4, P9			
T15	Quantitative Autonomy	P3, P8, P7, P9, P12	Σ		
T16	Trust Communities	P1, P5			
T17	Organic Capabilities C2 (Non-learning adaptive):	P3, P8, P4, P9	②		
T18	PLANETS	P3, P1			
T19	Organic Capabilities C1 (OC-ready)	P4	①		

Appendix I

Quality Function Deployment (QFD) Tables

Business Drivers vs Market Drivers

Code	Description	Fast economic development	Deeply connected	Hypercompetition	Standardization	Computerization	Knowledge intensive	**Score**	**Average Score**
M1	communication	8	6	9	8	2	3	36	6.0
M2	location flexibility	8	7	9	7	2	2	35	5.8
M3	automated decisions	6	6	6	5	3	4	30	5.0
M4	time flexibility	8	6	8	2	4	2	30	5.0
M5	adaptation	7	6	7	5	2	2	29	4.8
M6	trust	8	6	5	8	1	1	29	4.8
M7	cross-space coordination	6	7	3	3	4	5	28	4.7
M8	privacy	6	6	6	1	1	7	27	4.5
M9	computation	5	3	2	7	4	2	23	3.8
M10	lower cost	3	1	7	7	1	2	21	3.5

Market Drivers vs Product Drivers

		M5	M3	M1	M9	M7	M2	M10	M8	M4
Code	**Description**	adaptation	automated decisions	communication	computation	cross-space coordination	location flexibility	lower cost	privacy	time flexibility
	weight	4.8	5.0	6.0	3.8	4.7	5.8	3.5	4.5	5.0
P1	common simulation test-b	8	8	8	8	8	8	8	8	8
P2	standardization	7	7	9	5	8	8	8	3	8
P3	autonomic	8	8	3	8	8	8	8	3	9
P4	organic computing	9	8	7	7	7	6	7	3	6
P5	social awareness	5	8	3	5	7	7	6	2	7
P6	sticky information	2	3	6	2	3	8	6	9	3
P7	emergence	7	8	3	3	7	6	6	2	6
P8	equilibrium systems	8	7	4	5	6	6	7	2	4
P9	systems integration	2	2	5	2	8	8	8	5	6
P10	unbounded data	8	7	2	1	3	2	6	1	7
P11	hibernation	9	1	1	1	1	8	8	1	8
P12	hidden complexity	3	8	3	1	8	1	3	4	1

Product Drivers vs Technologies

		P3	P1	P7	P8	P11	P12	P4	P5	P2
Code	**Description**	autonomic	common simulation test-bed	emergence	equilibrium systems	hibernation	hidden complexity	organic computing	social awareness	standardization
	weight	6.5	8	5.1	5.1	3.9	3.6	6.3	5.6	6.6
T1	Quantitative Emergence	9	9	9	8	4	9	9	9	8
T2	Organic Capabilities C5 (Adaptive with offline rule generation and cooperation)	9	9	9	9	6	9	9	9	7
T3	Holonic Systems	6	9	9	8	6	9	9	8	6
T4	rules of enduring institutic	9	9	9	9	2	4	9	9	9
T5	Evolutionary simulation	6	9	9	8	3	1	8	8	8
T6	Organic Capabilities C4 (Adaptive with offline rule generation)	9	8	9	9	4	8	9	9	6
T7	Cloud Computing	7	9	7	7	1	3	7	7	7
T8	Swarm Robotics	8	8	7	7	1	8	8	8	6
T9	Organic Capabilities C3 (Adaptive with online learning):	9	8	8	9	3	7	9	9	5
T10	TIMIPLAN	7	8	4	4	8	8	5	5	7
T11	Open distributed systems	7	7	7	6	2	1	7	7	9
T12	data flow model	7	9	5	5	2	3	5	4	8
T13	Machine learning	6	9	6	6	2	4	6	6	3
T14	Organic Network Control	9	6	4	3	2	6	8	2	4

Appendix J

Timeline: 2014 | 2018 | 2022 | 2026 | 2030 | 2034 | 2038 | 2042 | 2046 | 2050

Phases: integration (2014–2030) | autonomous decisions (2030–2038) | discovery (2038–2050)

Group	Item	Scores
Business Drivers	(B1) Fast economic development	
	(B2) Hyper Competition	
	(B3) Deeply Connected	
	(B4) Standardization	
	(B5) Knowledge intensive	
	(B6) Computerization	
Market Drivers	(M1) communication	6.0
	(M2) location flexibility	5.8
	(M3) automated decisions	5.0
	(M4) time flexibility	5.0
	(M5) adaptation	4.8
	(M6) trust	4.8
	(M7) cross-space coordination	4.7
	(M8) privacy	4.5
	(M9) computation	3.8
	(M10) lower cost	3.5
Product Drivers	(P1) common simulation test-bed ⑤④③	384
	(P2) standardization	322
	(P3) autonomic ⑤④②	308
	(P4) organic computing ⑤④③ ②	303
	(P5) social awareness ⑤	269
	(P6) sticky information	249
	(P7) emergence ⑤④∑	246
	(P8) equilibrium systems ⑤④③ ∑②	243
	(P9) systems integration ⑤④③ ∑②	232
	(P10) unbounded data	190
	(P11) hibernation	187
	(P12) hidden complexity ⑤④③ ∑	172
Technologies	(T1) Quantitative Emergence	485
	⑤ (T2) Organic Capabilities C5 (Adaptive with offline rule generation and cooperation)	483
	(T3) Holonic Systems	474
	(T4) rules of enduring institutions	462
	(T5) Evolutionary simulation	449
	④ (T6) Organic Capabilities C4 (Adaptive with offline rule generation)	448
	(T7) Cloud Computing	430
	(T8) Swarm Robotics	420
	③ (T9) Organic Capabilities C3 (Adaptive with online learning):	420
	(T10) TIMIPLAN	418
	(T11) Open distributed systems	415
	(T12) data flow model	374
	(T13) Machine learning	332
	(T14) Organic Network Control	319
	∑(T15) Quantitative Autonomy	310
	(T16) Trust Communities	300
	② (T17) Organic Capabilities C2 (Non-learning adaptive):	272
	(T18) PLANETS	147
	① (T19) Organic Capabilities C1 (OC-ready)	110

References

Akidau, T., Bradshaw, R., Chambers, C., Chernyak, S., Fernández-Moctezuma, R. J., Lax, R., ... & Whittle, S. (2015). The dataflow model: a practical approach to balancing correctness, latency, and cost in massive-scale, unbounded, out-of-order data processing. *Proceedings of the VLDB Endowment*, *8*(12), 1792-1803.

Becerra-Fernandez, I., & Sabherwal, Rajiv. (2015). Knowledge management : Systems and processes (Second ed.). London ; New York: Routledge.

Chang, V. (2014). The Business Intelligence as a Service in the Cloud. Future Generation Computer Systems, 37(C), 512-534.

Cohen, W., & Levinthal, D. (1990). Absorptive Capacity: A New Perspective on Learning and Innovation. Administrative Science Quarterly, 35(1), 128-152.

Cong Vinh, P. (2016). Concurrency of self-* in autonomic systems. Future Generation Computer Systems, 56, 140-152.

Frey, & Osborne. (2017). The future of employment: How susceptible are jobs to computerisation? Technological Forecasting & Social Change, 114, 254-280.

Galvin, R. (2004). Roadmapping—A practitioner's update. Technological Forecasting & Social Change, 71(1), 101-103.

Gerdsri, N., Assakul, P., & Vatananan, R. (2010). An activity guideline for technology roadmapping implementation. Technology Analysis & Strategic Management, 22(2), 229-242.

Huang, Lin, Tsai, Yan, & Shih. (2018). Building edge intelligence for online activity recognition in service-oriented IoT systems. Future Generation Computer Systems, 87, 557-567.

Jansen, Justin J.P., Van Denbosch, Frons A.J., & Volberda, Henk W. (2005). Managing potential and realized absorptive capacity: How do organizational antecedents matter? Academy of Management Journal, 48(6), 999-1015.

Jiang, Guo, Liang, Lai, & Wen. (2017). Searchain: Blockchain-based private keyword search in decentralized storage. Future Generation Computer Systems, Future Generation Computer Systems.

Kephart, J., & Chess, D. (2003). The vision of autonomic computing. *Computer*, *36*(1), 41-50.

Li, Ping, Li, Tong, Ye, Heng, Li, Jin, Chen, Xiaofeng, & Xiang, Yang. (2018). Privacy-preserving machine learning with multiple data providers. Future Generation Computer Systems, 87, 341-350.

McCluskey, T., Kotsialos, Apostolos, Müller, J. P., Klügl, Franziska, Rana, Omer, & Schumann, René. (2016). *Autonomic road transport support systems* (Autonomic systems). Switzerland: Birkhäuser.

Muller-Schloer, Christian. Tomforde, Sven. (2018). *Organic Computing - Technical Systems for Survival In the Real World*. S.L.: Birkhauser.

Ostrom, E. (2011). Governing the commons: The evolution of institutions for collective action. Cambridge, Mass: Cambridge University Press.

Phaal, & Muller. (2009). An architectural framework for roadmapping: Towards visual strategy. Technological Forecasting & Social Change, 76(1), 39-49.

Prybila, Schulte, Hochreiner, & Weber. (2017). Runtime verification for business processes utilizing the Bitcoin blockchain. Future Generation Computer Systems, Future Generation Computer Systems.

Roman, Lopez, & Mambo. (2018). Mobile edge computing, Fog et al.: A survey and analysis of security threats and challenges. Future Generation Computer Systems, 78, 680-698.

Romanski, Bobby. (2017). The Evolution to an Autonomous Economy using the Blockchain. Engineering and Technology Management Student Projects.

Scholtes, P., Joiner, Brian L, & Streibel, Barbara J. (2003). The team handbook (3rd ed.). Madison, WI: Oriel.

Ship Map. (2018). Shipmap.org. [online] Available at: https://www.shipmap.org/ [Accessed 21 Sep. 2018].

Skorobogatov, S. (2010). Real world AES key extraction. Presentation, Rump Session, Santa Barbara, USA.

Standard Inputs for Eurocontrol Cost Benefit Analyses. (2012, February 24). States News Service, p. States News Service, Feb 24, 2012.

Von Hippel, E. (1994). "Sticky Information" and the Locus of Problem Solving: Implications for Innovation. Management Science, 40(4), 429-439.

Watson, I. (1997). Applying case-based reasoning: Techniques for enterprise systems. San Francisco, Calif.: Morgan Kaufmann.

Zhuge, H. (2014). Cyber-Physical Society—The science and engineering for future society. Future Generation Computer Systems, 32, 180-186.